Runbook for BrokerX

Overview

This runbook provides operational procedures and troubleshooting steps for managing BrokerX.

Starting with Docker

Starting with `deploy.sh` script

1. To start the service, begin by cloning the Github repository.
2. Run `BrokerX/brokerX/deploy.sh`
3. The application should now be running on port 8000

Starting manually

If you prefer not to use the script or it does not work, you can follow these steps.

1. To start the service, begin by cloning the Github repository.
2. Go to `BrokerX/brokerX`
3. Run `docker compose down -v` (or `docker compose down` if you want previous data to persist)
4. Run `docker compose build`
5. Run `docker compose up` (or `docker compose up -d` if you want it running in the background)
6. The application should now be running on port 8000

Using the VM

Note that the `deploy.sh` script is automatically called on the VM in the CD script. You can access the application at `http://10.194.32.208:8000/`.

Diagnosticating errors

Errors are automatically logged in the `django.error_logs` file, alternatively you can use the command `docker logs -f broker_app` or `docker logs -f broker_mysql` to access the docker logs.

Accessing the database

You can access the MySQL command line as root by running the following command: `docker exec -it broker_mysql mysql -u root -p`

Running tests

You can run the tests by running the command `docker exec broker_app python -m pytest`, or if you wish to get the coverage `docker exec broker_app python -m pytest --cov=broker --cov-config=.coveragerc --cov-report=term-missing`